|  |
| --- |
| **EX NO.:11 FILE ORGANIZATION TECHNIQUES**  **DATE:** |

**AIM:**

To implement the file organization techniques.

**1.SINGLE LEVEL:**

**CODING:**

#include<stdio.h>

struct

{

char dname[10],fname[10][10];

int fcnt;

}dir;

void main()

{

int i,ch;

char f[30];

clrscr();

dir.fcnt = 0;

printf("\nEnter name of directory -- ");

scanf("%s", dir.dname);

while(1)

{

printf("\n\n 1. Create File\t2. Delete File\t3. Search File \n 4. Display Files\t5. Exit\nEnter your choice -- ");

scanf("%d",&ch);

switch(ch)

{

case 1: printf("\n Enter the name of the file -- ");

scanf("%s",dir.fname[dir.fcnt]);

dir.fcnt++;

break;

case 2: printf("\n Enter the name of the file -- ");

scanf("%s",f);

for(i=0;i<dir.fcnt;i++)

{

if(strcmp(f, dir.fname[i])==0)

{

printf("File %s is deleted ",f);

strcpy(dir.fname[i],dir.fname[dir.fcnt-1]);

break;

}

}

if(i==dir.fcnt)

printf("File %s not found",f);

else

dir.fcnt--;

break;

case 3: printf("\n Enter the name of the file -- ");

scanf("%s",f);

for(i=0;i<dir.fcnt;i++)

{

if(strcmp(f, dir.fname[i])==0)

{

printf("File %s is found ", f);

break;

}

}

if(i==dir.fcnt)

printf("File %s not found",f);

break;

case 4: if(dir.fcnt==0)

printf("\n Directory Empty");

else

{

printf("\n The Files are -- ");

for(i=0;i<dir.fcnt;i++)

printf("\t%s",dir.fname[i]);

}

break;

default: exit(0);

}

}

getch();

}

**OUTPUT:**

Enter name of directory -- CSE

1. Create File 2. Delete File 3. Search File

4. Display Files 5. Exit Enter your choice – 1

Enter the name of the file -- A

1. Create File 2. Delete File 3. Search File

4. Display Files 5. Exit Enter your choice – 1

Enter the name of the file -- B

1. Create File 2. Delete File 3. Search File

4. Display Files 5. Exit Enter your choice – 1

Enter the name of the file -- C

1. Create File 2. Delete File 3. Search File

4. Display Files 5. Exit Enter your choice – 4

The Files are -- A B C

1. Create File 2. Delete File 3. Search File

4. Display Files 5. Exit Enter your choice – 3

Enter the name of the file – ABC

File ABC not found

1. Create File 2. Delete File 3. Search File

4. Display Files 5. Exit Enter your choice – 2

Enter the name of the file – B

File B is deleted

1. Create File 2. Delete File 3. Search File

4. Display Files 5. Exit Enter your choice – 5

**2.TWO LEVEL:**

**CODING:**

#include<stdio.h>

struct

{

char dname[10],fname[10][10];

int fcnt;

}dir[10];

void main()

{

int i,ch,dcnt,k;

char f[30], d[30];

clrscr();

dcnt=0;

while(1)

{

printf("\n\n 1. Create Directory\t 2. Create File\t 3. Delete File");

printf("\n 4. Search File \t \t 5. Display \t 6. Exit \t Enter your choice -- ");

scanf("%d",&ch);

switch(ch)

{

case 1: printf("\n Enter name of directory -- ");

scanf("%s", dir[dcnt].dname);

dir[dcnt].fcnt=0;

dcnt++;

printf("Directory created");

break;

case 2: printf("\n Enter name of the directory -- ");

scanf("%s",d);

for(i=0;i<dcnt;i++)

if(strcmp(d,dir[i].dname)==0)

{

printf("Enter name of the file -- ");

scanf("%s",dir[i].fname[dir[i].fcnt]);

dir[i].fcnt++;

printf("File created");

break;

}

if(i==dcnt)

printf("Directory %s not found",d);

break;

case 3: printf("\nEnter name of the directory -- ");

scanf("%s",d);

for(i=0;i<dcnt;i++)

{

if(strcmp(d,dir[i].dname)==0)

{

printf("Enter name of the file -- ");

scanf("%s",f);

for(k=0;k<dir[i].fcnt;k++)

{

if(strcmp(f, dir[i].fname[k])==0)

{

printf("File %s is deleted ",f);

dir[i].fcnt--;

strcpy(dir[i].fname[k],dir[i].fname[dir[i].fcnt]);

goto jmp;

}

}

printf("File %s not found",f);

goto jmp;

}

}

printf("Directory %s not found",d);

jmp : break;

case 4: printf("\nEnter name of the directory -- ");

scanf("%s",d);

for(i=0;i<dcnt;i++)

{

if(strcmp(d,dir[i].dname)==0)

{

printf("Enter the name of the file -- ");

scanf("%s",f);

for(k=0;k<dir[i].fcnt;k++)

{

if(strcmp(f, dir[i].fname[k])==0)

{

printf("File %s is found ",f);

goto jmp1;

}

}

printf("File %s not found",f);

goto jmp1;

}

}

printf("Directory %s not found",d);

jmp1: break;

case 5: if(dcnt==0)

printf("\nNo Directory's ");

else

{

printf("\nDirectory\tFiles");

for(i=0;i<dcnt;i++)

{

printf("\n%s\t\t",dir[i].dname);

for(k=0;k<dir[i].fcnt;k++)

printf("\t%s",dir[i].fname[k]);

}

}

break;

default:exit(0);

}

}

getch();

}

**OUTPUT:**

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit Enter your choice -- 1

Enter name of directory -- DIR1

Directory created

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit Enter your choice -- 1

Enter name of directory -- DIR2

Directory created

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit Enter your choice -- 2

Enter name of the directory – DIR1

Enter name of the file -- A1

File created

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit Enter your choice -- 2

Enter name of the directory – DIR1

Enter name of the file -- A2

File created

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit Enter your choice -- 2

Enter name of the directory – DIR2

Enter name of the file -- B1

File created

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit Enter your choice -- 5

Directory Files

DIR1 A1 A2

DIR2 B1

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit Enter your choice -- 4

Enter name of the directory – DIR

Directory not found

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit Enter your choice -- 3

Enter name of the directory – DIR1

Enter name of the file -- A2

File A2 is deleted

1. Create Directory 2. Create File 3. Delete File

4. Search File 5. Display 6. Exit Enter your choice – 6

**3.HIERARCHICAL LEVEL:**

**CODING:**

#include<stdio.h>

#include<graphics.h>

struct tree\_element

{

char name[20];

int x,y,ftype,lx,rx,nc,level;

struct tree\_element \*link[5];

};

typedef struct tree\_element

node; void main()

{

int gd=DETECT,gm;

node \*root;

root=NULL;

clrscr();

create(&root,0,"root",0,639,320);

clrscr();

initgraph(&gd,&gm,"c:\\tc\\BGI");

display(root);

getch();

closegraph();

}

create(node \*\*root,int lev,char \*dname,int lx,int rx,int x)

{

int i,gap;

if(\*root==NULL)

{

(\*root)=(node \*)malloc(sizeof(node));

printf("Enter name of dir/file(under %s) :",dname);

fflush(stdin);

gets((\*root)->name);

printf("enter 1 for Dir/2 forfile :");

scanf("%d",&(\*root)->ftype);

(\*root)->level=lev;

(\*root)->y=50+lev\*50;

(\*root)->x=x;

(\*root)->lx=lx;

(\*root)->rx=rx;

for(i=0;i<5;i++)

(\*root)->link[i]=NULL;

if((\*root)->ftype==1)

{

printf("No of sub directories/files(for %s):",(\*root)->name); scanf("%d",&(\*root)->nc);

if((\*root)->nc==0)

gap=rx-lx;

else gap=(rx-lx)/(\*root)->nc;

for(i=0;i<(\*root)->nc;i++)

create(&((\*root)->link[i]),lev+1,(\*root)->name,lx+gap\*i,lx+gap\*i+gap,lx+gap\*i+gap/2);

}

else (\*root)->nc=0;

}

}

display(node \*root)

{

int i;

settextstyle(2,0,4);

settextjustify(1,1);

setfillstyle(1,BLUE);

setcolor(14); if(root!=NULL)

{

for(i=0;i<root->nc;i++)

{

line(root->x,root->y,root->link[i]->x,root->link[i]->y);

}

if(root->ftype==1) bar3d(root->x-20,root->y-10,root->x+20,root->y+10,0,0); else

fillellipse(root->x,root->y,20,20);

outtextxy(root->x,root->y,root->name); for(i=0;i<root->nc;i++)

{

display(root->link[i]);

}

}

}

**OUTPUT:**

Enter Name of dir/file (under root): ROOT

Enter 1 for Dir / 2 For File : 1

No of subdirectories / files (for ROOT) :2

Enter Name of dir/file (under ROOT):USER 1

Enter 1 for Dir /2 for file:1

No of subdirectories /files (for USER 1):1

Enter Name of dir/file (under USER 1):SUBDIR

Enter 1 for Dir /2 for file:1

No of subdirectories /files (for SUBDIR):2

Enter Name of dir/file (under USER 1):

JAVA Enter 1 for Dir /2 for file:1

No of subdirectories /files (for JAVA): 0

Enter Name of dir/file (under SUBDIR):VB

Enter 1 for Dir /2 for file:1

No of subdirectories /files (for VB): 0

Enter Name of dir/file (under ROOT):USER2

Enter 1 for Dir /2 for file:1

No of subdirectories /files (for USER2):2

Enter Name of dir/file (under ROOT):A

Enter 1 for Dir /2 for file:2

Enter Name of dir/file (under USER2):SUBDIR 2

Enter 1 for Dir /2 for file:1

No of subdirectories /files (for SUBDIR 2):2

Enter Name of dir/file (under SUBDIR2):PPL

Enter 1 for Dir /2 for file:1

No of subdirectories /files (for PPL):2

Enter Name of dir/file (under PPL):B

Enter 1 for Dir /2 for file:2

Enter Name of dir/file (under PPL):C

Enter 1 for Dir /2 for file:2

Enter Name of dir/file (under SUBDIR):AI

Enter 1 for Dir /2 for file:1

No of subdirectories /files (for AI): 2

Enter Name of dir/file (under AI):D

Enter 1 for Dir /2 for file:2

Enter Name of dir/file (under AI):E

Enter 1 for Dir /2 for file:2

**RESULT:**

Thus the file organization techniques are studied and executed successfully.